Detecting Splicing and Copy-Move Attacks in Color Images

Mohammad Manzurul Islam[1], Gour Karmakar[1], Joarder Kamruzzaman[1], Manzur Murshed[1], Gayan Kahandawa[1] and Nahida Parvin[2]
[1]*School of Science, Engineering and Information Technology, Federation University Australia,* Melbourne, Australia
{mm.islam, gour.karmakar, joarder.kamruzzaman, manzur.murshed, g.appuhamillage}@federation.edu.au
[2]*Department of Computer Science and Engineering, Stamford University Bangladesh,* Dhaka, Bangladesh
nahida.cse@stamforduniversity.edu.bd

***Abstract*— Image sensors are generating limitless digital images every day. Image forgery like splicing and copy-move are very common type of attacks that are easy to execute using sophisticated photo editing tools. As a result, digital forensics has attracted much attention to identify such tampering on digital images. In this paper, a passive (blind) image tampering identification method based on Discrete Cosine Transformation (DCT) and Local Binary Pattern (LBP) has been proposed. First, the chroma components of an image is divided into fixed sized non-overlapping blocks and 2D block DCT is applied to identify the changes due to forgery in local frequency distribution of the image. Then a texture descriptor, LBP is applied on the magnitude component of the 2D-DCT array to enhance the artifacts introduced by the tampering operation. The resulting LBP image is again divided into non-overlapping blocks. Finally, summations of corresponding inter-cell values of all the LBP blocks are computed and arranged as a feature vector. These features are fed into a Support Vector Machine (SVM) with Radial Basis Function (RBF) as kernel to distinguish forged images from authentic ones. The proposed method has been experimented extensively on three publicly available well-known image splicing and copy-move detection benchmark datasets of color images. Results demonstrate the superiority of the proposed method over recently proposed state-of-the-art approaches in terms of well accepted performance metrics such as accuracy, area under ROC curve and others.**

Keywords*— *Digital forensics, splicing attack, copy-move attack, chroma components, Discrete Cosine Transformation, Local Binary Pattern, Support Vector Machine.

I. INTRODUCTION

Internet of Things (IoT) objects have been an integrated part in today's digital life. According to Ericsson Mobility Report on June 2018 [1], there will be 31.4 billion connected devices by 2023, which turns out to be approximately four devices per person breathing on planet earth. We are connecting our everyday smart objects such as smart vehicles, home appliances, security devices, wearable sensors, visual sensors, etc. to the Internet. A large number of sensors among those are visual sensors which play an important role in the cyberspace as well as physical security and surveillance. Again, modern social media like Instagram, Snapchat and Facebook as well as electronic news media are generating millions of digital images every day. People tend to rely and trust on these image data generated by these visual sensors and online media. However, specialized digital photo editing software and tools have made it quite easy to tamper images and thus anyone can easily generate fake images that look very much natural and authentic. Human visual system cannot identify any trace of forgery within these images. Spreading negative propaganda, hiding the actual facts, confusion in decision making process have become more common in modern media as a result of image forgery. Among all possible image forgery techniques, splicing and copy-move are the most well-known and commonly used attacks on images [2]. In image splicing, one or more portions of an image are copied and then pasted on another image. On the other hand, in copy-move attack, one or more portions of an image are copied and then pasted on a different part of the same image.

A single image can describe a complex idea easily. Therefore, an artificially tampered image may give a totally opposite or different impression and produce quite disturbing consequences [3, 4]. Photojournalist Markus Schreiber took the picture in Fig. 1(a) on the very first day of the 2017 G-20 summit, Germany [5]. Later this picture was modified and uploaded to social media by a Russian journalist [6], which created a huge rumour and confusion across the world. Similarly, a tampered image could affect the decisions that are mainly based on digital image information.

(a) Authentic image (b) Spliced image

Fig. 1. Image splicing example

Although digital image forgery does not leave any clue to the naked eyes, they can be detected through digital forensics. There exist different methodologies in literature for identifying image tampering. They can be grouped into two broad categories: active [7, 8] and passive (blind) [9, 10]. Active methods implant a unique signature or digital watermark into the source image. It is done mostly by the image sensor or camera while capturing the image. The receiver of the image verifies its authenticity by checking the signature or watermark. But majority of the image sensors available for consumers are cost effective and run on low resources for which they are not so capable of applying digital signature and watermarking techniques [11, 12]. Therefore, active methods have limited applications. However, passive (blind) approaches do not require such prior knowledge and hence attract much attention in digital image forgery detection in recent years. The key idea behind the blind detection approach is that image splicing and copy-move attacks introduce such artifacts which cannot be identified by human eyes but still they leave some footprints by altering statistical and structural properties of the image. To be precise, splicing and copy-move attacks introduces unexpected changes and micro-patterns along the pasted region boundary. We can

consider these tampering artifacts as noise inserted into clear signal and thus identify them.

Even though there are many image tamper detection techniques available in the current literature, further improvements in accuracy is needed for reliable forgery detection. In this paper, we propose a passive detection method using discrete cosine transformation (DCT) and local binary pattern (LBP) for detecting splicing and copy-move tampering from the chroma channels of an input image. Firstly, we identify the tampering artifacts in frequency domain by applying 2-D block DCT on image chroma channels. Secondly, LBP operator is used to enhance these artifacts. In order to propagate the changes introduced by image forgery, we extract the features from the blocks of LBP image by summing up the relevant inter-cell values. Finally, to classify the images as authentic or tampered, these features are fed into a support vector machine (SVM). Our proposed method outperforms many popular and contemporary methods in terms of classification accuracy produced by three benchmark datasets. This is mainly for the use of the new features and the order of the application of DCT and LBP.

The rest of the paper is organized as follows. Section II reports the related works. Section III explains our proposed method in detail. Section IV discusses the experimental result of our proposed method. It also compares our method with existing methods. Finally, Section V concludes the article.

II. Related Works

The rapid growth of camera technology and image sensors are the reason for billions of images being generated every month. Consequently, image forgery has also been increased dramatically. Researchers are continuously making efforts on detecting image forgery and thus they have proposed different approaches in recent years. In the following subsection, we mention some of the promising methods proposed by different researchers. All of them differ mainly based on the approach they adopt to model the structural and statistical changes in forged images. Most of the methods reported below utilized SVM for learning and classification.

Johnson and Farid [13] proposed a method for identifying tampered image using the inconsistency in lighting conditions within the image. But their method does not work well if the host image and the image from where the forged portion is taken are under similar lighting conditions. In [14], Hsu and Chang proposed a method for splicing detection using geometry invariants and camera response function. However, their method is semi-automatic because of user engagement in labelling suspicious region in the tampered image, which is unrealistic in real-time applications. Later, they improved the method by integrating automatic segmentation [15].

Researchers in [11, 16, 17] suggested run-length based techniques to detect image forgery. Dong et al. [16] used a run-length and edge statistics to detect splicing attacks with 76.52% accuracy for Columbia gray dataset [18]. Later on, He et al. [17] improved the accuracy (80.58%), computational cost and reduced the feature dimensionality of this method. In [11], Zhao et al. proposed a method where features were extracted from de-correlated chroma channels and four gray level run-length run-number (RLRN) vectors with different directions. They found that RLRN performed better in chroma space than in RGB or luminance space for detecting image splicing. An accuracy of 85% on Columbia color dataset [14] and 94.7% on CASIA 1 dataset [19] were reported.

Shi et al. [20] introduced a method using statistical moment features and Markov features extracted from both pixel domain and frequency (multi-block DCT coefficients) domain. They improved the moment features that have been already used for steganalysis [21] in their previous research. He et al. [22] enhanced this method by extracting the Markov features from both DCT and DWT frequency domains. Unlike [20], they considered both intra-block and inter-block correlation among DCT coefficients and also discarded all moment based features. The main problem of these two methods ([20] and [22]) is that their detection accuracies (84.86% and 89.76%) are not good enough for CASIA 2 dataset [23], a more challenging dataset in nature [19]. Wang et al. [24] proposed a method by modelling the edge information of an image in chroma space as a finite-state Markov chain and used its stationary distribution as features. They achieved higher detection accuracy (95.6%) for CASIA 2 dataset than that of [20] and [22].

Some researchers utilized texture descriptors like Weber Local Descriptor (WLD) and LBP to model image tampering artifacts. In [25], Hussain et al. compared multiscale WLD with multiscale LBP. They achieved better result using WLD (94.29% for splicing, 90.97% for copy-move) than using LBP (90.48% splicing and 85.83% for copy-move) on CASIA 1 dataset. In another work by Hussain et al. [26], features were extracted using multi-resolution WLD from the chroma component of images and reported 93.33% detection accuracy for splicing attacks and 91.52% detection accuracy for copy-move attacks over CASIA 1. Muhammad et al. [27] applied steerable pyramid transform (STP) on the chroma components of image and then calculated LBP histogram to generate features. They achieved 94.89%, 97.33% and 96.39% detection accuracy on CASIA 1, CASIA 2 and Columbia color dataset respectively.

Zhang et al. [28] and Alahmadi et al. [12] used DCT and LBP, however the order of DCT and LBP application on image blocks, the color space of image and feature extraction techniques are different. Zhang et al. [28] applied LBP operator on the magnitude of 2D-DCT coefficient of the gray image blocks, and used the histogram to generate features. In contrast, Alahmadi et al. [12] applied LBP operator on non-overlapping blocks of chrominance channels followed by 2D-DCT transformation. They used the standard deviation based features calculated using the corresponding DCT coefficients. The accuracy of both methods appears to be promising. To be precise, Alahmadi et al. found better detection accuracy using chrominance channel of an image. Many methods (e.g., [11, 12, 29-31]) support the effectiveness of chrominance channels over illuminance channel for detecting image forgery artifacts. Inspired by the ability of chroma channels along with the efficacy of DCT and LBP operators to capture splicing and copy-move attacks, we propose a splicing and copy-move detection method using DCT, LBP and a different feature set calculated by an aggregation operator for color images. The proposed method is presented in the following section.

III. Proposed Method

Detecting splicing and copy-move attacks are binary decision problem – whether an image is altered or not. The

forgery introduces both statistical and structural changes in an original image. It affects the extracted features of an image. Therefore, a number of techniques are required to apply on the image before deriving final features. These features are then fed into a classifier to identify fake images. The overall mechanism of the proposed method has been illustrated in Fig. 2. In the following sections, we discuss our proposed method in detail.

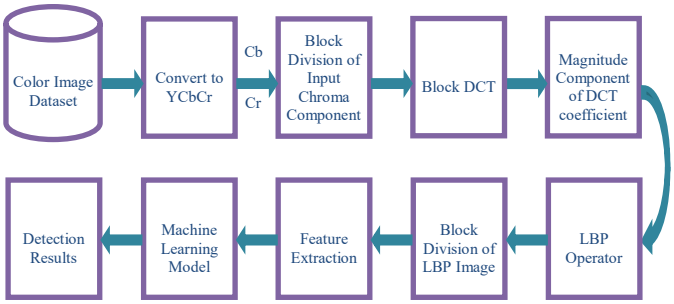

Fig. 2. Proposed image splicing and copy-move detection system

A. *Converting images into YCbCr color space to extract Cb, Cr component*

We have implemented our system using three publicly available image splicing and copy-move detection datasets. All the images in the datasets are in RGB color space. Chroma components hold most of the tampering artifacts that human eyes cannot perceive. Therefore, we first convert them into YCbCr color space to extract the Cb and Cr components. Here, Y refers to luminance component while Cb is the difference between the blue component and a reference value, and Cr is the difference between the red component and a reference value [32, 33]. We used the following transformation formula to convert an RGB image into YCbCr color image [34].

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (1)$$

B. *Block division of input Chroma Component*

An image can be manipulated in different ways by splicing and copy-move attacks. Different image segments with different sizes may have been pasted on different part of the host image. In addition, the host image itself can be of different dimensions. As a result, different sized block division of the image is essential to correctly capture the forgery artifacts. In our proposed method, we divide the images into blocks in two phases and discuss the first phase in this section while the second phase is discussed in Section III-E. In the first phase, we divide the chroma components (Cb and Cr) of the YCbCr color image into different block sizes. We have experimented our proposed method with different sizes of blocks: 4x4, 8x8 and 16x16 as well as combining features from all these blocks. We divide the chroma channels into blocks using the following procedure. Let $C^{wb\times hb}$ be a Cb or Cr channel of the input image of size $wb \times hb$ pixels. We divide $C^{wb\times hb}$ into $w \times h$ non-overlapping blocks of size $b \times b$ pixels. The resultant chroma component block 2D array is given by,

$$C^{wb\times hb} = \begin{bmatrix} C_{1,1}^{b\times b} & \cdots & C_{1,w}^{b\times b} \\ \vdots & \ddots & \vdots \\ C_{h,1}^{b\times b} & \cdots & C_{h,w}^{b\times b} \end{bmatrix}. \quad (2)$$

C. *Block discrete cosine transformation (BDCT)*

Because of image forgery, unexpected changes and micro patterns are introduced along the tampered regions of an image. The splicing and copy-move attack disturbs the natural correlation between image pixels by altering regularity, smoothness, continuity of the tampered image and by changing the local frequency distribution of the affected areas [20]. It is also necessary to condense the diversity of the image content and enhance the tampering artifacts before extracting final features. To represent the degree of content change in a chroma component, it is converted from pixel domain to frequency domain using BDCT. BDCT shows encouraging performance in representing pixel domain changes in local frequency distribution because of its remarkable capability of pixel decorrelation and energy compaction properties [35]. We apply 2D-DCT on the blocks of $C^{wb\times hb}$ to generate DCT coefficients. Let $Y^{wb\times hb}$ be the resultant transform domain coefficient after applying 2D-DCT on each block and it is given by,

$$Y^{wb\times hb} = \begin{bmatrix} Y_{1,1}^{b\times b} & \cdots & Y_{1,w}^{b\times b} \\ \vdots & \ddots & \vdots \\ Y_{h,1}^{b\times b} & \cdots & Y_{h,w}^{b\times b} \end{bmatrix}, \quad (3)$$

where, $Y_{i,j}^{b\times b} = 2D\text{-}DCT\left(C_{i,j}^{b\times b}\right), 1 \le i \le w, 1 \le j \le h$. The 2D-DCT of an input block $C_{i,j}^{b\times b}$ and output block $Y_{i,j}^{b\times b}$ is given by,

$$Y_{i,j}^{b\times b}(p,q) = \alpha_p \alpha_q \sum_{m=0}^{b-1} \sum_{n=0}^{b-1} C_{i,j}^{b\times b}(m,n) \cos\frac{\pi(2m+1)p}{2b} \cos\frac{\pi(2n+1)q}{2b}, \quad (4)$$

where, $0 \le p \le b-1, 0 \le q \le b-1$ and

$$\alpha_p = \begin{cases} \sqrt{\frac{1}{b}}, & \text{if } p = 0 \\ \sqrt{\frac{2}{b}}, & \text{otherwise} \end{cases}, \quad (5)$$

$$\alpha_q = \begin{cases} \sqrt{\frac{1}{b}}, & \text{if } q = 0 \\ \sqrt{\frac{2}{b}}, & \text{otherwise} \end{cases}. \quad (6)$$

D. *Local binary pattern (LBP) operator*

We employ LBP operator on the magnitude component of $Y^{wb\times hb}$ in order to identify and enhance different splicing and copy-move artifacts. LBP is a robust and computationally inexpensive texture descriptor. We adopt LBP in our system mainly to identify the occurrences of micro-patterns in the forged images. LBP highlights these forgery artifacts and augment them in the host image. In LBP, every pixel value of a given 2D array is compared with its neighbouring pixel values and an LBP code is generated for that pixel. Note, in

our proposed system, we apply LBP operator on the DCT coefficients of an image and hence it is calculated as below:

Let $L^{wb\times hb}$ be the resultant LBP array generated by applying LBP operator on magnitude components of $Y^{wb\times hb}$. It is given by,

$$L^{wb\times hb} = LBP_{N,R}(|Y^{wb\times hb}|)\,, \tag{7}$$

$$LBP_{N,R} = \sum_{n=0}^{N-1} g(p_n - p_c)2^n\,. \tag{8}$$

where N, R and p_c represent the number of neighbor DCT coefficients, the radius and the central DCT coefficient, respectively. The neighbouring DCT coefficient is defined as p_n where $n = 0,1,\dots,N-1$. The function $g(p_n - p_c)$ is defined as,

$$g(p_n - p_c) = \begin{cases} 1, & p_n - p_c \geq 0 \\ 0, & p_n - p_c < 0 \end{cases}. \tag{9}$$

In our proposed system, we use the basic LBP where the values are calculated in a rectangular window. Here, we choose $N = 8$ and $R = 1$. Therefore, the central DCT coefficient p_c compares its own value with neighbouring 8 DCT coefficients. If the central DCT coefficient's value is equal or smaller than the neighbouring DCT coefficient's value, then 1 is recorded; otherwise 0 is recorded. Based on these assessments, central DCT coefficient p_c generates an 8-bit binary code and converts it to a decimal number to stores as its own LBP code. We explain the process with an example in Figure 3. Here, we take a sample segment from the magnitude component of $Y^{wb\times hb}$ and calculate LBP. The binary values are obtained by comparing the central DCT coefficient p_c with its 8 neighboring DCT coefficients. Then the 8-bit binary digit is formed starting from Least Significant Bit (LSB) to Most Significant Bit (MSB). Finally, the binary number is converted to decimal number and LBP code replaces the DCT coefficient value of central DCT coefficient p_c.

Fig. 3. LBP code generation procedure

E. Block division of LBP image

This is the second and final phase of block division where we divide the LBP 2D array $L^{wb\times hb}$ into same size of blocks, similar to the block division operation performed in Section III-B. We divide $L^{wb\times hb}$ into $w \times h$ non-overlapping blocks of size $b \times b$ LBP codes. The resultant block 2D array of LBP is represented as,

$$L^{wb\times hb} = \begin{bmatrix} L_{1,1}^{b\times b} & \cdots & L_{1,w}^{b\times b} \\ \vdots & \ddots & \vdots \\ L_{h,1}^{b\times b} & \cdots & L_{h,w}^{b\times b} \end{bmatrix}. \tag{10}$$

F. Feature extraction

DCT has the capabilities of representing changes among pixel values of an input chroma component by transforming it from spatial domain to frequency domain. A forged image may have higher and more non-zero values of high frequency DCT coefficients depending on the degree of change in that image. In the proposed method, the features are extracted by calculating the summation of corresponding inter-cell values of blocks of LBP codes. We adopt such approach in this specific order because, DCT coefficient capture the changes among pixel values in spatial domain by transforming it to frequency domain, while LBP enhances these changes and thus magnifies the forgery artifacts in higher frequency components. We need to preserve these local changes captured by LBP to make the forgery detection system efficient as much as possible. These local changes can be regarded as outlier since splicing and copy-move attacks usually introduce sophisticated changes in images. It is well known that mean is most affected by outliers than other statistical measures. The feature extraction technique in our proposed method is based on an aggregation operator (sum) which is similar to the mean based feature extraction technique. We implemented our proposed method with different block sizes as discussed in Section III-B and Section III-E. The features in our proposed method are derived as below:

Let $Z_k^{w\times h}$ be the k-th LBP code values of all blocks in $L^{wb\times hb}$. Therefore,

$$Z_k^{w\times h} = \begin{bmatrix} L_{1,1}^{b\times b}(k) & \cdots & L_{1,w}^{b\times b}(k) \\ \vdots & \ddots & \vdots \\ L_{h,1}^{b\times b}(k) & \cdots & L_{h,w}^{b\times b}(k) \end{bmatrix},\ 1 \leq k \leq b^2\,, \tag{11}$$

where, $L_{u,v}^{b\times b}(k)$ is the k-th LBP code of that block. The k-th feature F_k is derived as,

$$F_k = \sum_{u=1}^{w}\sum_{v=1}^{h} L_{u,v}^{b\times b}(k)\,. \tag{12}$$

IV. Experiments and Results

A. Description of datasets

For the evaluation of our proposed method, we have used three standard benchmark datasets for image splicing and copy-move attack detection: (i) Columbia Uncompressed Image Splicing Detection Evaluation Dataset (Columbia Color) [14], (ii) CASIA Tampered Image Detection Evaluation Database version 1.0 (CASIA TIDE v1.0) [19] and (iii) CASIA Tampered Image Detection Evaluation Database version 2.0 (CASIA TIDE v2.0) [23]. Among them, CASIA 2 is the latest dataset that contains high resolution and color images of different types and resolutions. We agree with [22-24] that CASIA 2 is the most realistic publicly available dataset having a large number of samples and thus, a robust splicing detection system should have higher accuracy on this dataset. The detailed information of these three datasets is shown in Table I.

TABLE I. INFORMATION OF THE DATASETS USED IN EVALUATION

Dataset	Image Size	Image Type	No. of Images			Tampering Method
			Authentic	*Tampered*	*Total*	
Columbia Color	757 x 568 - 1152 x 768	TIF, BMP	183	180	363	Simple crop-and-paste using Photoshop, No post processing
CASIA 1	384 x 256, 256 x 384	JPG	800	921	1721	Photoshop with pre-processing; No post-processing
CASIA 2	240 x 160 - 900 x 600	JPG, TIF, BMP	7491	5123	12614	Photoshop with pre-processing and/or post-processing

B. SVM Classifier and model validation

In machine learning, SVM has been widely used as it shows promising performance in many well recognized applications of splicing and copy-move attack detection. We chose LIBSVM [36] as the classifier to evaluate the accuracy of the proposed system. Radial Basis Function (RBF) kernel was selected for this work. The regularisation parameter (c) and variance of RBF kernel (γ) were identified through a 'loose and fine' grid-search method [37]. The performance of our method was evaluated using sixfold cross-validation. All the classification related tasks were performed in Weka 3.8.2 [38]. We used MATLAB R2017b for feature extraction and data pre-processing related tasks.

C. Results and discussion

The detection accuracy for features derived from Cb component, Cr component and their combination has been summarized in Table II. All of them were tested for block size of 4x4, 8x8, 16x16 as well as their combination (4x4 + 8x8 + 16x16). The detection accuracy increases along with an increase of block size. Therefore, use of 4x4 blocks produces the lowest accuracy, while combining the blocks results in the highest accuracy across different datasets. In terms of chroma components, we found the best result using Cb component in Columbia color dataset, while the best results were obtained when we combined both Cb and Cr component in CASIA 1 and CASIA 2 datasets. The variations of detection accuracy in different chroma channels are observed in different datasets because of the nature of images [12]. For example, images in Columbia dataset were mostly taken in indoor conditions with exactly four specific models of cameras, whereas CASIA 1 and CASIA 2 contain images that were taken both indoor and outdoor environment using different sources (e.g., Corel image dataset, websites, own camera sources). The detection accuracy using Cb component, Cr component and their combination for each dataset varies between 1% ~ 2% approximately (Table II) which indicates that our proposed system is quite robust. Choosing either of the chroma components can provide satisfactory outcome. Overall, the proposed method achieves detection accuracy of 97.52%, 97.79% and 99.82% over Columbia color, CASIA 1 and CASIA 2 dataset respectively. In addition to detection accuracy, we also determined precision, recall and AUC (Area Under ROC curve) of our method using Weka tool which are reported in Table II.

D. Comparison with recent methods

There are other methods for detecting splicing and copy-move attacks found in existing literature as described in Section II. Among them, two existing methods ([28] and [12]) adopted both DCT and LBP in their systems and reported good detection accuracy. Zhang et al. [28] proposed their system using gray scale images while Alahmadi et al. [12] used color images. Since our proposed method is for detecting splicing and copy-move attacks in color images, we have implemented the latter one to compare thoroughly with our proposed system. Table III presents the comparison of our proposed method with Alahmadi et al.'s method using the combined Cb and Cr components for all three datasets mentioned in Section IV-A. The basic experimental setup remains the same as mentioned in Section IV-B.

False Negative Rate (FNR) and False Positive Rate (FPR) are the two significant performance evaluation metrics for any image forgery detection system where FNR refers to the rate of erroneously classifying a tampered image as an authentic one (i.e., miss rate) and FPR means the rate of falsely classifying an authentic image as a tampered one which generates false alarm. From the forensic or security viewpoint, missing a tampered image has more severe consequences and thus a reliable detection system must produce as low FNR as possible. To assess the robustness and validity of our forgery detection method, the FNR should be significantly lower than existing methods. Table III shows that our proposed method produces 1.83% ~ 1.11% lower FNR than the existing method in [12]. In addition, our method performs better in terms of True Positive Rate (TPR), True Negative Rate (TNR) and AUC. For all three datasets, our method achieves better AUC ([0.975, 0.977, 0.998] vs [0.967, 0.970, 0.976]), which is a more accepted metric considering performance in both classes. From Table III, it is clear that our method outperforms the method in [12] throughout all three datasets.

TABLE II. OVERALL DETECTION ACCURACY IN OUR PROPOSED METHOD WITH VARYING BLOCK SIZE AND DIFFERENT CHROMA COMPONENTS (CB, CR AND COMBINED CB+CR)

Block Size	Evaluation	Columbia Color			CASIA 1			CASIA 2		
		Cb	*Cr*	*Cb+Cr*	*Cb*	*Cr*	*Cb+Cr*	*Cb*	*Cr*	*Cb+Cr*
4x4	***Accuracy***	92.287	89.532	93.113	83.217	86.247	87.995	99.207	99.215	99.247
	Precision	0.952	0.928	0.938	0.830	0.860	0.866	0.988	0.989	0.989
	Recall (TPR)	0.889	0.856	0.922	0.863	0.888	0.917	0.993	0.992	0.993
	AUC	0.923	0.895	0.931	0.830	0.861	0.877	0.992	0.992	0.993
8x8	***Accuracy***	96.143	94.490	95.041	94.173	95.047	95.571	99.429	99.350	99.516
	Precision	0.988	0.971	0.982	0.935	0.944	0.950	0.992	0.991	0.992
	Recall (TPR)	0.933	0.917	0.917	0.958	0.965	0.968	0.994	0.993	0.996
	AUC	0.961	0.945	0.950	0.941	0.949	0.955	0.994	0.993	0.995
16x16	***Accuracy***	97.245	94.490	95.317	95.047	96.911	97.028	99.715	99.715	99.730
	Precision	0.978	0.994	0.977	0.936	0.966	0.962	0.996	0.995	0.996
	Recall (TPR)	0.967	0.894	0.928	0.974	0.977	0.984	0.997	0.998	0.998
	AUC	0.972	0.944	0.953	0.949	0.968	0.969	0.997	0.997	0.997
4x4 + 8x8 + 16x16	***Accuracy***	**97.521**	94.766	95.592	96.154	97.727	**97.786**	99.786	99.786	**99.818**
	Precision	0.983	0.988	0.961	0.953	0.976	0.973	0.997	0.996	0.997
	Recall (TPR)	0.967	0.906	0.950	0.976	0.982	0.986	0.998	0.998	0.998
	AUC	0.975	0.947	0.956	0.960	0.977	0.977	0.998	0.998	0.998

TABLE III. COMPARISON BETWEEN PROPOSED METHOD AND METHOD IN [12]

Dataset	Evaluation	Proposed Method	Method in [12]
Columbia Color	***Accuracy (%)***	97.521	96.694
	FPR (%)	1.64	2.19
	FNR (%)	3.33	4.44
	TPR (%)	96.67	95.56
	TNR (%)	98.36	97.81
	AUC	0.975	0.967
CASIA 1	***Accuracy (%)***	97.786	96.969
	FPR (%)	3.14	3.01
	FNR (%)	1.41	3.05
	TPR (%)	98.59	96.95
	TNR (%)	96.86	96.99
	AUC	0.977	0.970
CASIA 2	***Accuracy (%)***	99.818	97.494
	FPR (%)	0.20	2.86
	FNR (%)	0.16	1.99
	TPR (%)	99.84	98.01
	TNR (%)	99.80	97.14
	AUC	0.998	0.976

In addition to comparing our method with Alahmadi et al.'s method, Table IV depicts the comparison of detection accuracy among different existing methods across different datasets.

TABLE IV. COMPARISON OF DETECTION ACCURACY OF PROPOSED METHOD WITH OTHER EXISTING METHODS AS REPORTED BY AUTHORS

Method \ Dataset	Columbia Color	CASIA 1	CASIA 2
	Accuracy (%)		
Proposed Method	97.52	97.79	99.82
Alahmadi et al. [12]	96.69	96.97	97.49
Muhammad et al. [27]	96.39	94.89	97.33
Hussain et al. [25]	94.29	-	-
Zhao et al. [11]	85.00	94.70	-
He et al. [22]	-	-	89.76
Shi et al. [20][a]	-	-	84.86
Zhang et al. [28][b]	91.38	-	-

[a.] implemented by [22]
[b.] implemented by [12]

Not all works have experimented with all three datasets mentioned above and therefore, here we only report results on the specific dataset(s) they reported. Table IV shows that our method outperforms existing state-of-the-art methods in all three benchmark datasets. To the best of our knowledge, the proposed method has achieved splicing and copy-move detection accuracy of up to 99.82%, the highest accuracy among all other methods available in the literature.

V. CONCLUSION

In this paper, a robust model has been proposed for detecting splicing and copy-move attacks in color images using DCT and LBP operator. Chrominance components are affected more by these attacks than luminance component. DCT is used to capture the change in the local frequency distribution, while LBP is applied on the magnitude component of the DCT coefficients to detect the occurrences of micro-patterns and magnify the artifacts introduced by splicing and copy-move attacks. Finally, summation of relevant inter-cell LBP values are calculated to extract features. We used SVM with RBF kernel to classify the images into authentic and tampered ones. The detection results show that the proposed method is superior to the existing methods across different well known publicly available benchmark datasets for image forgery detection.

ACKNOWLEDGMENT

Credits for the use of the Columbia Image Splicing Detection Evaluation Dataset are given to the DVMM Laboratory of Columbia University, CalPhotos Digital Library and the photographers listed in https://bit.ly/2B4ckLt. Also, credits for the use of the CASIA Image Tempering Detection Evaluation Database (CAISA TIDE) V1.0 and V2.0 are given to the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Corel Image Database and the corresponding photographers.

REFERENCES

[1] P. Jonsson and S. Carson, "Ericsson mobility report," Ericsson, Stockholm, Sweden, 2018.

[2] J. A. Redi, W. Taktak, and J.-L. Dugelay, "Digital image forensics: a booklet for beginners," *Multimedia Tools and Applications,* vol. 51, no. 1, pp. 133-162, 2011.

[3] A. Mallonee, "Infamously altered photos, before and after their edits," Wired, USA, 2015, Available: https://bit.ly/2Ia8zqf.

[4] "List of photo manipulation controversies." wikipedia.org, 2018, Available: https://bit.ly/2wcweBB

[5] M. Schreiber, "APTOPIX Trump Germany G20," ed: Associated Press, 2017.

[6] M. Novak, "That viral photo of Putin and Trump is totally fake," gizmodo.com, 2017, Available: https://bit.ly/2JR9VmN.

[7] R. Kwitt, P. Meerwald, and A. Uhl, "Lightweight detection of additive watermarking in the DWT-domain," *IEEE Transactions on Image Processing,* vol. 20, no. 2, pp. 474-484, 2011.

[8] V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *3rd IEEE International Conference on Industrial Informatics*, 2005, pp. 709-716.

[9] B. Mahdian and S. Saic, "A bibliography on blind methods for identifying image forgery," *Signal Processing: Image Communication,* vol. 25, no. 6, pp. 389-399, 2010.

[10] G. K. Birajdar and V. H. Mankar, "Digital image forgery detection using passive techniques: A survey," *Digital Investigation,* vol. 10, no. 3, pp. 226-245, 2013.

[11] X. Zhao, J. Li, S. Li, and S. Wang, "Detecting digital image splicing in chroma spaces," in *IWDW 2010: Digital Watermarking,* Berlin, Heidelberg, pp. 12-22.

[12] A. A. Alahmadi, M. Hussain, H. A. Aboalsamh, M. Ghulam, G. Bebis, and H. Mathkour, "Passive detection of image forgery using DCT and local binary pattern," *Signal, Image and Video Processing,* vol. 11, pp. 81-88, 2017.

[13] M. K. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Transactions on Information Forensics and Security,* vol. 2, no. 3, pp. 450-461, 2007.

[14] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *2006 IEEE Internation Conference on Multimedia and Expo*, Canada, pp. 549-552.

[15] Y. F. Hsu and S. F. Chang, "Image splicing detection using camera response function consistency and automatic segmentation," in *2007 IEEE International Conference on Multimedia and Expo*, pp. 28-31.

[16] J. Dong, W. Wang, T. Tan, and Y. Q. Shi, "Run-length and edge statistics based approach for image splicing detection," in *IWDW 2008: Digital Watermarking*, Berlin, Heidelberg, pp. 76-87.

[17] Z. He, W. Sun, W. Lu, and H. Lu, "Digital image splicing detection based on approximate run length," *Pattern Recognition Letters,* vol. 32, no. 12, pp. 1591-1597, 2011.

[18] T.-T. Ng and S.-F. Chang, "A model for image splicing," in *IEEE International Conference on Image Processing*, Singapore, 2004, pp. 1169-1172.

[19] J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," in *IEEE International Conference on Signal and Information Processing*, 2013, pp. 422-426.

[20] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in *Proceedings of the 9th workshop on Multimedia and security*, USA, 2007, pp. 51-62.

[21] Y. Q. Shi, "Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network," in *IEEE International Conference on Multimedia and Expo*, 2005, pp. 269-272.

[22] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern Recognition,* vol. 45, no. 12, pp. 4292-4299, 2012.

[23] J. Dong and W. Wang. "CASIA Tampered Imaged Detection Evaluation Database (CASIA TIDE v2.0)" [Online]. Available: http://forensics.idealtest.org/casiav2/

[24] W. Wang, J. Dong, and T. Tan, "Image tampering detection based on stationary distribution of Markov chain," in *IEEE International Conference on Image Processing*, 2010, pp. 2101-2104.

[25] M. Hussain, S. Q. Saleh, H. Aboalsamh, G. Muhammad, and G. Bebis, "Comparison between WLD and LBP descriptors for non-intrusive image forgery detection," in *2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings*, pp. 197-204.

[26] M. Hussain, G. Muhammad, S. Q. Saleh, A. M. Mirza, and G. Bebis, "Image forgery detection using multi-resolution Weber local descriptors," in *Eurocon 2013*, pp. 1570-1577.

[27] G. Muhammad, M. H. Al-Hammadi, M. Hussain, and G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," *Machine Vision and Applications,* vol. 25, no. 4, pp. 985-995, 2014.

[28] Y. Zhang, C. Zhao, Y. Pi, S. Li, and S. Wang, "Image-splicing forgery detection based on local binary patterns of DCT coefficients," *Security and Communication Network,* vol. 8, no. 14, pp. 2386-2395, 2015.

[29] A. A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, and G. Bebis, "Splicing image forgery detection based on DCT and Local Binary Pattern," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 253-256.

[30] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *8th workshop on Multimedia and security*, Geneva, Switzerland, 2006, pp. 48-55.

[31] W. Wei, J. Dong, and T. Tan, "Effective image splicing detection based on image chroma," in *16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 1257-1260.

[32] C. A. Poynton, *A technical introduction to digital video*. John Wiley & Sons, Inc., 1996.

[33] G. C. Karmakar, "An integrated fuzzy rule-based image segmentation framework," PhD disseration, Monash University, Victoria, Australia, 2002.

[34] R. C. Gonzalez, *Digital Image processing using MATLAB*. Upper Saddle River, N. J.: Pearson Prentice Hall, 2004.

[35] S. A. Khayam, "The discrete cosine transform (DCT): theory and application," *Michigan State University,* vol. 114, 2003.

[36] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology,* vol. 2, pp. 27:1-27:27, 2011.

[37] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Department of Computer Science, National Taiwan University, 2016.

[38] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online appendix for "Data mining: Practical machine learning tools and techniques"*, Fourth Edition ed. Morgan Kaufmann, 2016.